Rozwiązanie testu

25 czerwca 2015

1. $\mathsf{SAT} \in \mathsf{NC}$. nie wiadomo

   Nie wiemy nawet, czy SAT nie należy do NC^1.

2. $\mathsf{BPP} \cap \mathsf{coBPP} = \mathsf{ZPP}$. nie wiadomo

   Klasa BPP jest zamknięta na dopełnienie, więc oczywiście $\mathsf{BPP} \cap \mathsf{coBPP} = \mathsf{BPP}$, podczas gdy $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$. Stąd oczywiście $\mathsf{ZPP} \subseteq \mathsf{BPP}$, ale przeciwna inkluzja nie jest znana.

3. Dla każdego problemu NP-zupełnego istnieje PTAS. nie wiadomo

   Wprawdzie wiadomo, że dla *problemu komiwojażera* nie istnieje algorytm aproksymacyjny (a więc tym bardziej PTAS), ale **przy założeniu**, że $\mathsf{P} \neq \mathsf{NP}$. Gdyby $\mathsf{P} = \mathsf{NP}$, to każdy optymalizacyjny problem w NP miałby algorytm deterministyczny, który w trywialny sposób spełnia założenia PTAS (błąd = 0).

4. $\mathsf{NPSPACE} \cap \mathsf{coNPSPACE} = \mathsf{PSPACE}$. tak

   Deterministyczna klasa PSPACE jest oczywiście zamknięta na dopełnienie, a z Tw. Savitcha $\mathsf{NPSPACE} = \mathsf{PSPACE}$. Zatem wszystkie występujące tu klasy są równe.

5. $\mathsf{NP} \cap \mathsf{coNP} = \mathsf{P}$. nie wiadomo

6. $\mathsf{EXPSPACE} = \mathsf{coEXPSPACE}$. tak

   Z definicji złożoności pamięciowej.

7. $\mathsf{P} = \mathsf{P/poly}$. nie

   Klasa P/poly jest klasą złożoności niejednorodnej i zawiera nieprzeliczalnie wiele problemów, w szczególności problemy nieobliczalne.

8. $\mathsf{AC} = \mathsf{NC}$. tak

   Z definicji obu klas $\mathsf{AC} = \bigcup \mathsf{AC}^n$, $\mathsf{NC} = \bigcup \mathsf{NC}^n$ oraz inkluzji $\mathsf{NC}^n \subseteq \mathsf{AC}^n \subseteq \mathsf{NC}^{n+1}$.

9. $\mathsf{BPP} \subseteq \mathsf{NP}$. nie wiadomo

   Wiemy, że $\mathsf{BPP} \subseteq \mathsf{PSPACE}$, więc brak inkluzji implikowałby $\mathsf{NP} \neq \mathsf{PSPACE}$ (co jest znanym problemem otwartym). Z drugiej strony, gdyby inkluzja zachodziła, możliwe byłyby dwa przypadki. (1) Pewien problem NP-zupełny L jest w BPP. Ale ta klasa jest zamknięta na dopełnienia, więc również $\overline{L}$ jest w BPP, co pociąga za sobą $\mathsf{coNP} \subseteq \mathsf{NP}$ (inny problem otwarty). (2) W przeciwnym razie inkluzja jest ostra, a ponieważ $\mathsf{P} \subseteq \mathsf{BPP}$, mielibyśmy $\mathsf{P} \neq \mathsf{NP}$.

10. $\mathsf{PCP}(\mathcal{O}(1), \mathcal{O}(1)) = \mathsf{P}$. tak

    Wielomianowa maszyna używająca stałej liczby bitów losowych i pytająca o stałą liczbę bitów dowodu może być łatwo zasymulowana przez maszynę deterministyczną.

11. $\mathsf{IP} = \mathsf{coIP}$.

    Z Tw. Shamira $\mathsf{IP} = \mathsf{PSPACE}$, a ta ostatnia klasa jest, jak już zauważyliśmy, zamknięta na dopełnienie.

12. Jeśli $\mathsf{P} = \mathsf{NP}$, to $\mathsf{EXPTIME} = \mathsf{NEXPTIME}$. tak

    Technika tzw. podkładania poduszek (ang. *padding*). Przypuśćmy, że L jest rozpoznawany przez niedeterministyczną maszynę w czasie c^{n^k}; wtedy język $\{w\perp^{c^{|w|^k}} : w \in L\}$ jest rozpoznawany przez niedeterministyczną maszynę w czasie $\mathcal{O}(n)$ (gdzie $\perp$ jest nowym symbolem). Hipotetyczny *deterministyczny* algorytm wielomianowy dla tego ostatniego języka dałby się łatwo przełożyć na deterministyczny algorytm wykładniczy dla języka L, w którym pierwsza faza polegałaby na dopisaniu odpowiedniej liczby $\perp$.

13. $\mathsf{AC}^0 \subseteq \mathsf{Reg}$, gdzie Reg to klasa języków regularnych. nie

    Łatwo jest skonstruować obwód o stałej głębokości rozpoznający palindromy.

14. $\mathsf{Reg} \subseteq \mathsf{AC}^0$. nie

    Wiadomo z wykładu, że język $PARITY$, czyli język regularny $0^*\left(10^*10^*\right)^*$ nie należy do AC^0.

15. Klasa $\mathsf{DTIME}(2^n)$ jest zamknięta na redukcje w $\mathsf{LOGSPACE}$. nie

    Znowu *padding*. Z Tw. o hierarchii istnieje język L w $\mathsf{DTIME}(2^{n^2}) - \mathsf{DTIME}(2^n)$. Wtedy język $\{w\perp^{|w|^2} : w \in L\}$ jest w $\mathsf{DTIME}(2^n)$. Jednak nietrudno jest (w $\mathsf{LOGSPACE}$) zredukować język L do tego ostatniego języka.